ЗАДАЧИ ЗА ПРОГРАМИРАНЕ НА ИТЕРАТИВНИ ПРОЦЕСИ

Румяна Жекова

МГ 'Баба Тонка', Русе 7000, ул. Иван Вазов 18
rngekova@abv.bg

Една интересна група са алгоритмите за изчисляване по итеративна или подобна формула. За да се разбере по-добре идеята им, е подходящо в часовете по програмиране да се разгледат повече задачи и то като отделна тема. Така работата по тях може да се „облече" в модел и времето за усвояването им да се намали. За целта предлагам някои примерни задачи, взети от помощните помагала, както и методиката, която съм използвала за по-лека работа в час.

Ключови думи: *редица, итерация, брой, точност*

Една интересна група са алгоритмите за изчисляване по итерационна или подобна формула. Обикновено за усвояването на идеята в час се използва много време, въпреки че (може би защото) предлаганите задачи в помощните помагала са малко. Моето предложение е пред учениците да се формулира отделна тема и в нея да се разгледат отделните стъпки и различните начини за изчисление.

Първа група може да включва задачите за намиране на итеративна сума, като:

- в началото се предложи на учениците да работят с краен брой събираеми (броят на събираемите може да се въведе от клавиатурата)

- след това се пристъпва към формулиране на обща итерационна задача за намиране на безкрайна сума.

- се започне с формули, които са реализирани в повечето алгоритмични езици, за да могат учениците сами да сравнят резултата си с търсената стойност и да разгледат процеса на сходимост.

Задача 1'. Да се изчисли сумата по дадени X и N.

$$S = 1 + \frac{X}{1!} + \frac{X^2}{2!} + \frac{X^3}{3!} + \ldots + \frac{X^N}{N!}$$

Задачата съдържа сума и интересни събираеми, номерирани с числата от 1 до N (предполага цикъл FOR). Всяко от събираемите може да се получи от предходното и това е желателно, защото, като цяло, числителят и знаменателят от някой момент нататък ще получат доста големи стойности и ще се наложи компютърът да закръглява няколко пъти (за числителя, за знаменателя и за

цялото събираемо). Второто неудобство е, че при отделните числител и знаменател, на някого може да му хрумне идеята да организира самостоятелни цикли за събираемото A_K, X^K и K! в общия цикъл за сумата (ненужно усложняване). А как от събираемото A_{K-1}, което е записано в клетката с име A, да се получи следващото събираемо A_K.

$$\frac{X^{K-1}}{(K-1)!}\cdot\frac{X}{K}=\frac{X^K}{K!}$$

$$A_{K-1} * X / K = A_K$$

$$A := A * X / K$$

Вече се оформя цикъл с поне две команди в тялото си.

```
{ Pascal 7.0 }
FOR K := 1 TO N DO BEGIN
A := A * X / K;
S := S + A          END;
```

```
// C++
for (int k = 1; k <= n ; k++)
{a = a * x  /  k;
s = s + k; }
```

Следващият съществен момент е задаването на началните стойности на двете клетки, в които се натрупват съответно събираемо и сума. След като се установи, че в началото те трябва да имат стойности 1, желателно е да се провери дали формулите, които са изведени за получаване на новото събираемо и сумата са напълно верни (особено за K равно на 1, 2, 3 и N). По-добре е в такива случаи X да не се замества с конкретна стойност, а съдържанието на клетките да се записва като изрази.

Ако ще се наблюдава процеса на сходимост, в тялото на цикъла се поставя оператор за извеждане на стойностите, които ще се сравняват. Тук може да се каже, че компютърът знае за тази сума и с нея изчислява E^X като функция с името exp(X).

```
Program zad1_1; { Pascal 7.0 }
Var K , N : Word; X , S , A : Real;
Begin
  Readln ( X , N );
  A := 1 ; S : = 1 ;
  For K := 1 To N Do Begin
    A := A * X / K;
    S := S + A ;
    Writeln ( Exp(X):18:9 );
    Writeln ( S:18:9)
  End;
End.
```

```
#include <iostream.h> // C++
#include <math.h>
viod main()
{
int n; double x, s, a;
  cin << x << n;
  a = 1; s = 1;
  for (int k = 1; k <= n ; k++)
  {a = a * x  /  k;
   s = s + k;
   cout <<"\n" <<exp(x) <<"\n" <<s;}
}
```

В крайния вариант на програмата, операторът за печат ще бъде извън цикъла и ще съдържа единствено клетката S. Естествено е след анализ да се стигне в час до извода, че такава сума се изчислява с немалък брой събираеми. При това броят на събираемите зависи от стойността на X, което е и естествено. Все пак не всеки трябва да се интересува от вида на формулата, а само от крайния резултат. Затова в реалните задачи броят на събираемите не се задава, а краят на изчисленията се достига при достатъчно близък резултат. **В ЧАС Е ДОСТАТЪЧНО** да се обясни **САМО**, че сумата ще се изчислява докато разликата от две нейни поредни стойности стане по-малка от Eps – достатъчно малко число /от ранга на 1E-6/. Имайки предвид, че в задачата разликата между S_{K-1} и S_K е току що прибавеното събираемо, което може да е положително или отрицателно число, то условието за край на цикъла е Abs(A)<Eps.

Цикълът вече е с постусловие, но в него се използва стойността на променливата K и затова програмата трябва да се погрижи за нея – начална стойност и стъпка, като вече не е задължително клетката-брояч да е целочислена.

Задача 1”. Пресметнете с точност Eps по формулата:

$$E^X = 1 + \frac{X}{1!} + \frac{X^2}{2!} + \frac{X^3}{3!} + \ldots + \frac{X^N}{N!} + \ldots$$

```
Program zad1_2; { Pascal 7.0 }
Const Eps=1E-6;
Var K : Word; X , S , A : Real;
Begin
Readln ( X );
A := 1 ; S : = 1 ; K : = 1 ;
Repeat
A := A * X / K;
S := S + A ;
K := K + 1
Until Abs (A) < Eps;
Writeln ( S:18:9 )
End.
```

```
#include <iostream.h> // C++
#include <math.h>
viod main()
{ const eps=1e-6;
 double x, s, a;
  cin << x ;
  a = 1; s = 1; k=1;
  do
  {a = a * x  /  k;    s = s + k;
   k++}
while abs(a) <eps;
cout <<“\n” <<s;}
}
```

Задача 2. Пресметнете с точност Eps по формулата:

$$\sin X = \frac{X}{1!} - \frac{X^3}{3!} + \frac{X^5}{5!} - \frac{X^7}{7!} + - \ldots$$

Задача 3. Пресметнете с точност Eps по формулата:

$$\cos X = 1 - \frac{X^2}{2!} + \frac{X^4}{4!} - \frac{X^6}{6!} + - \ldots$$

Тези две задачи са изключително близки помежду си, тъй като тялото на цикъла им се оказва едно и също и се отнасят за функции, които съществуват в алгоритмичните езици (Забележка: аргументът е в радиани, а не в градуси). За тях също ще е необходима малко помощ, защото малко ученици се досещат лесно, че знакът на една стойност се променя най-леко с унарната операция минус.

Следващата задача е съвсем лека след първите три и е особена единствено с липсата на аргумент и факта, че вдясно от знака за равенство се получава само четвърт от стойността на числото П.

Задача 4. Пресметнете с точност Eps числото П по формулата на Готфрид Лайбниц от 1673 година:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + - \ldots$$

Задача 5. Пресметнете с точност Eps числото П по формулата на Шарп от 1699 година:

$$\pi = 2.\sqrt{3}.\left(1 - \frac{1}{3^2.3} - \frac{1}{3^2.5} - \frac{1}{3^2.7} - \ldots\right)$$

Задача 6. Пресметнете с точност Eps числото П по формулата на Ойлер от 1736 година:

$$\pi^2 = 6.\left(1 + \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \ldots\right)$$

Задача 7. Пресметнете с точност Eps по формулата:

$$F(X) = 1 + \frac{X}{2} - \frac{X^2}{2!} - \frac{X^3}{2.2!} + \frac{X^4}{3!} + \frac{X^5}{2.3!} - \ldots$$

Задача 7 е особена с факта, че събираемите са групирани по две. Това не е проблем, но усложнява изчисляването на формулата за следващото събираемо и проверката накрая.

Задача 8. Пресметнете с точност Eps по формулата:

$$F(x) = \frac{X}{1!} + \frac{1}{2}.\frac{X^3}{3!} + \frac{1.3}{2.4}.\frac{X^5}{5!} + \frac{1.3.5}{2.4.6}.\frac{X^7}{7!} + \ldots$$

Задача 9. Пресметнете с точност Eps по формулата:

$$Ln(x) = 2.\left(Y + \frac{Y^3}{3} + \frac{Y^5}{5} + \frac{Y^7}{7} + \ldots \right), \text{където } Y = \frac{X-1}{X+1}$$

Задача 10. Пресметнете с точност Eps по формулата:

$$Acrtg(x) = X - \frac{X^3}{3!} + \frac{X^5}{5!} - \frac{X^7}{7!} + \ldots, \text{където} -1 < X < 1$$

Задачи 9 и 10 обясняват вградени функции и отново дават възможност за сравнение и самопроверка на крайния резултат.

Задача 11. Пресметнете с точност Eps по формулата за периметър на елипса с полуоси A и B:

$$U = 2.\pi.A\left(1 - \frac{1}{2}.e^2 + \frac{1.3}{2.4}.\frac{e^4}{3} + \frac{1.3.5}{2.4.6}.\frac{e^6}{5} + \ldots \right), \text{където } A \geq B \text{ и } e^2 = 1 - \left(\frac{B}{A}\right)^2$$

Задача 11 е реално използваема, макар и доста рядко.

Задача 12. Пресметнете с точност Eps по формулата:

$$S = \frac{1+1!}{1} - \frac{2+2!}{1+1/2} + \frac{3+3!}{1+1/2+1/3} - \ldots + (-1)^{N+1}\frac{N+N!}{1+1/2+1/3+\ldots+1/N}$$

Втората група, от която предлагам три задачи, демонстрира, че не винаги итерационните формули представляват някакъв вид суми. При тях се налага да се запази старата стойност на търсения резултат, за да може да се сравни с новата и чрез тях да се провери условието за край на цикъла.

Задача 13.Пресметнете с точност Eps числото П по формулата на Уелс от 1656 година:

$$\pi = \frac{2.2}{1.3}.\frac{4.4}{3.5}.\frac{6.6}{5.7}. \quad \ldots \quad .\frac{(2k).(2k)}{(2k-1).(2k+1)}$$

Задача 14. Пресметнете $\sqrt[N]{X}$ с точност Eps като използвате зависимостта:

$$Y_{K+1} = \frac{1}{N}.\left(\frac{X}{Y_K^{N-1}} + (N-1).Y_K \right); \quad Y_1 = \frac{X}{N}$$

Задача 14. Пресметнете $\sqrt[N]{X}$ с точност Eps като използвате зависимостта:

$$Y_{K+1} = Y_K.\frac{(N-1).Y_K^N + (N+1).X}{(N+1).Y_K^N + (N-1).X}; \quad Y_1 = \frac{X}{N}$$

Литература:

1. Азълов П., Златарова Ф., Сборник информатика с Паскал в примери, тестове и задачи, София, Просвета, 1996 г

2. Василев Цв., Христова Пл., Марков М., Григорова К., Ръководство за упражнения по Програмиране и използване на компютрите – II част, Русе, Русенски университет "Ангел Кънчев", 2005 г

3. Тодорова М., Програмиране на C++ - част 1., София, СИЕЛА, 2004 г